

## CLAIMS

What is claimed is:

- 1     1.     A method, comprising:  
2             vectoring an instruction pointer to a firmware-based exception filter in  
3     response to an exception;  
4             executing the firmware-based exception filter; and  
5             re-vectoring the instruction pointer to an operating system (OS) exception  
6     handler configured to handle the exception.
  
- 1     2.     The method of claim 1, wherein execution of the firmware-based exception  
2     filter performs operations including saving at least one processor register value to a  
3     storage device.
  
- 1     3.     The method of claim 1, wherein execution of the firmware-based exception  
2     filter performs operations including saving at least a portion of system memory to a  
3     storage device.
  
- 1     4.     The method of claim 1, further comprising:  
2             loading a set of OS exception handler pointers into a first memory address  
3     space;  
4             relocating the set of OS exception handler pointers to a second memory  
5     address space; and  
6             loading a set of firmware-based exception filter pointers into the first address  
7     space.

1     5.     The method of claim 4, further comprising:  
2             storing a base address of the second memory address space; and  
3             employing the base address of the second memory address space to re-  
4     vector the instruction pointer to an OS exception handler pointer to the OS exception  
5     handler configured to handle the exception.

1     6.     The method of claim 1, further comprising:  
2             loading a set of OS exception handlers into a first memory address space;  
3             relocating the set of OS exception handlers to a second memory address  
4     space; and  
5             loading a set of firmware-based exception filters into the first address space.

1     7.     The method of claim 6, further comprising:  
2             storing a base address of the second memory address space; and  
3             employing the base address of the second memory address space to re-  
4     vector the instruction pointer to the OS exception handler configured to handle the  
5     exception.

1     8.     The method of claim 1, further comprising:  
2             loading a set of OS exception handler pointers into a first memory address  
3     space;  
4             setting a processor exception vector register to include a base address of the  
5     first memory address space;  
6             loading a set of firmware-based exception filter pointers into a second  
7     address space; and

8 replacing the base address of the first memory address space with the base  
9 address of the second memory address space in the processor exception vector  
10 register.

1 9 The method of claim 8, further comprising:  
2 storing a base address of the first memory address space; and  
3 employing the base address of the first memory address space to re-vector  
4 the instruction pointer to an OS exception handler pointer to the OS exception  
5 handler configured to handle the exception.

1 10. The method of claim 1, further comprising:  
2 loading a set of OS exception handlers into a first memory address space;  
3 setting a processor exception vector register to include a base address of the  
4 first memory address space;  
5 loading a set of firmware-based exception filters into a second address space;  
6 and  
7 resetting the processor exception vector register to include a base address of  
8 the second memory address space;

1 11. The method of claim 10, further comprising:  
2 storing a base address of the first memory address space; and  
3 employing the base address of the first memory address space to re-vector  
4 the instruction pointer to the OS exception handler configured to handle the  
5 exception.

1 12. The method of claim 1, further comprising:  
2 loading the firmware-based exception filter into system memory; and

3           fixing up code in the firmware-based exception filter to re-vector the  
4   instruction pointer to one of the OS exception handler configured to handle the  
5   exception or a pointer to the OS exception handler configured to handler the  
6   exception.

1   13.    A method, comprising:  
2           loading a set of operating system (OS)-based exception handler components  
3   into system memory;  
4           physically or logically replacing the set of OS-based exception handler  
5   components with a corresponding set of firmware-based exception filter and/or  
6   handler components;  
7           vectoring an instruction pointer to a firmware-based exception filter and/or  
8   handler in response to an OS runtime exception; and  
9           executing the firmware-based exception filter and/or handler.

1   14.    The method of claim 13, further comprising re-vectoring the instruction pointer  
2   to an operating system (OS) exception handler configured to handle the OS run-time  
3   exception after the firmware-based exception filter and/or handler has been  
4   executed.

1   15.    The method of claim 14, further comprising fixing up code in the firmware-  
2   based exception filter and/or handler to re-vector the instruction pointer to one of the  
3   OS exception handler configured to handle the OS runtime exception or a pointer to  
4   the OS exception handler configured to handle the OS runtime exception.

1   16.    The method of claim 13, wherein the set of OS-based exception handlers are  
2   physically replaced by:

3 copying the set of OS-based exception handlers from a physical address  
4 space to a virtual address space; and  
5 overwriting the physical address space with the set of firmware-based  
6 exception filter and/or handler components.

1 17. The method of claim 13, wherein the set of OS-based exception handlers are  
2 logically replaced by:

3 loading the set of OS-based exception handlers into a first memory address  
4 space having a first base address; and

5 loading the set of firmware-based exception filter and/or handler components  
6 into a second address space having a second base address; and

7 replacing the first base address with the second base address in a register  
8 that is used to locate the base address of a table containing one of a set of  
9 exception handler procedures or pointers to a set of exception handler procedures.

1 18. A machine-readable medium to provide instructions, which when executed  
2 perform operations including:

3 determining a first base address of a set of operating system (OS)-based  
4 exception handler components that have been loaded into a first memory address  
5 space;

6 storing the first base address;

7 loading a set of firmware-based exception filter and/or handler components  
8 into a second memory address space having a second base address; and

9 setting an exception vector register to have a base address corresponding to  
10 the second base address.

1 19. The machine-readable medium of claim 18, further to provide the set of  
2 firmware-based exception filter and/or handler components.

1 20. The machine-readable medium of claim 18, wherein the medium comprises a  
2 firmware storage device.

1 21. The machine-readable medium of claim 18, to provide further instructions to  
2 perform operations including:  
3 filtering a runtime exception using a firmware-based exception filter; and  
4 re-vectoring an instruction pointer to an operating system (OS) exception  
5 handler configured to handle the runtime exception.

1 22. A machine-readable medium to provide instructions, which when executed  
2 perform operations including:  
3 moving a set of operating system (OS)-based exception handler components  
4 from a first memory address space having a first base address to a second memory  
5 address space having a second base address;  
6 storing the second base address; and  
7 loading a set of firmware-based exception filter and/or handler components  
8 into the first memory address space.

1 23. The machine-readable medium of claim 22, further to provide the set of  
2 firmware-based exception filter and/or handler components.

1 24. The machine-readable medium of claim 22, wherein the medium comprises a  
2 firmware storage device.

1    25.    The machine-readable medium of claim 22, to provide further instructions to  
2    perform operations including:  
3            filtering a runtime exception using a firmware-based exception filter; and  
4            re-vectoring an instruction pointer to an operating system exception handler  
5    configured to handle the runtime exception.

1    26.    A system, comprising:  
2            a processor;  
3            memory, coupled to the processor;  
4            a flash device, having firmware instructions stored thereon to perform  
5    operations in combination with logic programmed into the processor, the operations  
6    including:  
7            loading a firmware-based exception filter into memory;  
8            detecting a runtime exception;  
9            vectoring an instruction pointer to the firmware-based exception filter in  
10    response to the runtime exception;  
11            executing the firmware-based exception filter; and  
12            re-vectoring the instruction pointer to an operating system (OS)  
13    exception handler configured to handle the runtime exception.

1    27.    The system of claim 26, further comprising a network interface coupled to the  
2    processor, wherein execution of firmware instructions loads a firmware-based  
3    exception filter from a network storage device via the network interface into the  
4    memory.

1    28.    The system of claim 26, wherein execution of the firmware instructions  
2    performs further operations including:

3       determining a first base address of a set of OS-based exception handler  
4 components that have been loaded into a first address space of the memory;  
5       storing the first base address;  
6       loading a set of firmware-based exception filter and/or handler components  
7 into a second address space of the memory having a second base address; and  
8       setting an exception vector register in the processor to have a base address  
9 corresponding to the second base address.

1   29.   The system of claim 26, wherein execution of the firmware instructions  
2 perform the further operation of fixing up code in the firmware-based exception filter  
3 to re-vector the instruction pointer to one of the OS exception handler configured to  
4 handle the runtime exception or a pointer to the OS exception handler configured to  
5 handler the runtime exception.

1   30.   The system of claim 26, wherein execution of the firmware instructions  
2 performs further operations including:  
3       moving a set of OS-based exception handler components from a first address  
4 space in the memory having a first base address to a second address space in the  
5 memory having a second base address;  
6       storing the second base address; and  
7       loading a set of firmware-based exception filter and/or handler components  
8 into the first memory address space.